

OVirt large guest feature

NUMA and Virtual NUMA

Version 0.91

Jason Liao

chuan.liao@hp.com

Background

Enhance oVirt, allow Enterprise customers to provision large guests for their traditional scale-up enterprise workloads and expect low overhead due to virtualization.

Feature 1: NUMA Node Tuning

Ability from the UI (with appropriate backend support) to specify the host NUMA node information for the backing memory of a large guest (i.e. via numatune with mode set to: strict, preferred or interleave) across specified host NUMA nodes. Here is an example from a XML config file of a guest where the backing guest's memory is interleaved between host NUMA nodes 0, 1.

```
<numatune>  
<memory mode='interleave' nodeset='0-1' />  
</numatune>
```

Ability from the UI to specify pin the VCPUs of a guest to the host's processors based on topology of the host socket/cores/threads.

Note: Automatic NUMA Balancing changes in the Linux kernel (i.e. upstream 3.13 kernel) should help reduce the need for having to explicitly specify this for a guest. But there will still be specific use cases where having this ability in the UI will prove useful.

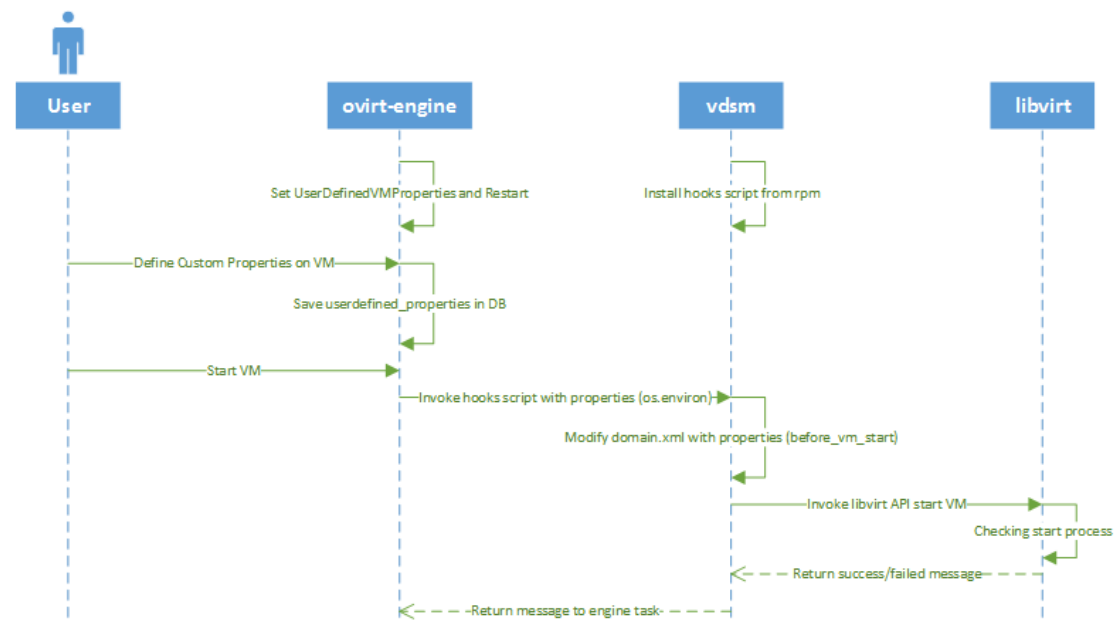
Feature 2: Guest NUMA topology

Ability from the UI (with the appropriate backend support) to specify and expose virtual NUMA nodes in a guest that spans more than a single host node. This allows the OS instance in the guest to take NUMA aware decisions and this improves scaling/performance within the guest. Here is an example from a guest XML config file where there are two virtual NUMA nodes in the guest.

```
<cpu>  
  <numa>  
    <cell cpus='0-7' memory='10485760' />  
    <cell cpus='8-15' memory='10485760' />  
  </numa>  
</cpu>
```

Proposal A: Using vdsd hooks script (before_vm_start)

Workflow



Design

1. Change ovirt-engine UserDefinedVMProperties
engine-config -s UserDefinedVMProperties=" "
cputune =[\w:\$]+;
numatune=[\w:\$]+;
guestnuma =[\w:\$]+;
2. Enter the custom properties with format in GUI

High Availability

Resource Allocation

Boot Options

Custom Properties

numa

strict:0

+

-

Name	Format
cputune	< vcpu >#<cpuset>,<vcpu2>#<cpuset2>
numatune	<memory policy>#<numaset>
guestnuma	<cellrange>#<memory>,<cellrange2>#<memory2>

3. Create the vdsd hooks script to change domain xml documents as libvirt virsh usage format for feature.
vdsd/vdsd_hooks/largeguest/before_vm_start

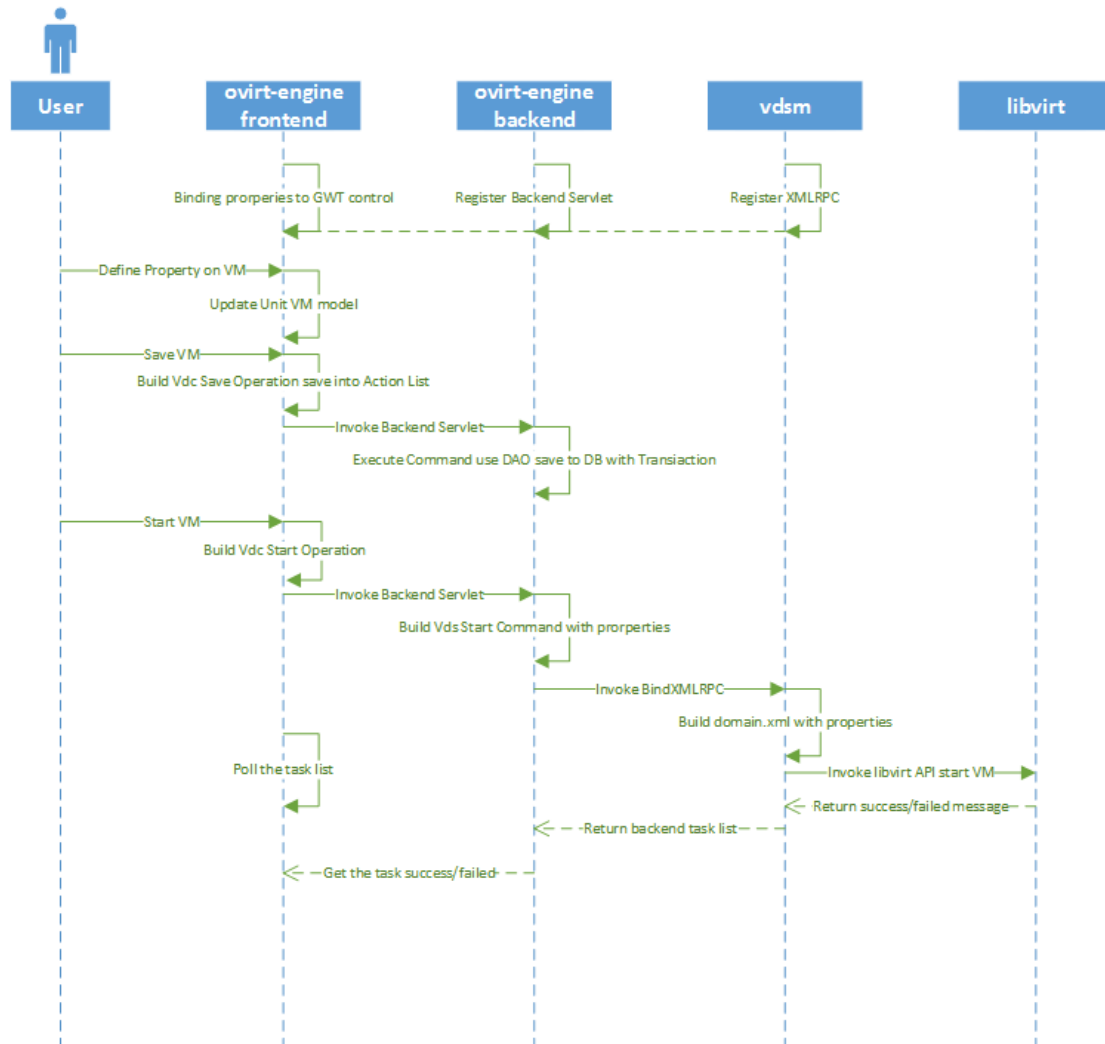
4. Create the related Makefile.am and README for rpm build and installation.
vdsd/vdsd_hooks/largeguest/Makefile.am
vdsd/vdsd_hooks/largeguest/README

Consideration

- vdsd/vdsd_hooks/numa/before_vm_start have the same feature of numatune
- vdsd/vdsd_hooks/pincpu/before_vm_start have the less feature of cputune
The script can only set one virtual CPU and host CPU mapping
- User could install the hook script on host from rpm package.
- User could use the GUI custom properties configuration large guest feature.
- The code change can be quickly implement on ovirt-node side.

Proposal B: Integrate GUI interface

Workflow



Design

1. Add three properties on VM define NUMA and Virtual NUMA

Name	Type	Comments
numaAffinity	Integer	0-no,1-auto,2-use numaTune
numaTune	String	<memory policy>#<numaset>
numaVirtual	String	<cellrange>#<memory>,<cellrange2>#<memory2>

Note:

- numaAffinity default is 0 (no affinity)
- memory policy default is strict
- numaset is the number of NUMA node split by comma
- cellrange is like 0-3 etc. memory is 512000 (512M) set by user

2. Add one properties on Host define NUMA node

Name	Type	Comments
numaNode	Integer	Node number

Note:

- This property is used for Add/Edit VM to display the host NUMA node.
- The host capabilities with the NUMA info is from vdsom interface.

3. Change GUI interface Add/Edit VM dialog with Host tab

The screenshot shows the 'New Virtual Machine' dialog box with the 'Host' tab selected. The 'Start Running On' section has 'Any Host in Cluster' selected. The 'Migration Options' section has 'Allow manual and automatic migration' selected. The 'CPU Pinning topology' section is expanded, showing 'NUMA Memory Affinity' with three radio buttons: 'No affinity', 'Auto affinity (numad)', and 'Use memory from nodes:'. The 'vNUMA / topology' section is also visible. A red box highlights the 'NUMA Memory Affinity' section.

Note:

- Only choose Start Running on specific host will open the red range configuration. As the same as CPU pinning.
- If the target host OS has support for Automatic NUMA balancing(numad) then the user can choose Auto affinity either use it or not.
- If the user chooses not to use it on a given host then they can disable it and choose to explicitly specify the host NUMA nodes for the backing memory of a guest.
- Choose Use memory from nodes will display the specific host NUMA info.

4. Change Frontend VM Module and related operation

- Modify [gwt-common](#) uibinder add two Textbox field
AbstractVmPopupWidget.ui.xml
- Modify [gwt-common](#) popup widget add two String properties

- AbstractVmPopupWidget.java
 - Modify [uicommonweb](#) vm module add two String properties
UnitVmModel.java
 - Modify [uicommonweb](#) vm list model new vm and on save action
VmListModel.java
 - Modify [uicommonweb](#) vm behavior add two properties validation
NewVMModelBehavior.java
ExistingVmModelBehavior.java
VmModleBehaviorBase.java
 - Change other localization files
5. Change Backend VM Module and related operation
- Modify [common](#) vm business entities add two String properties
Host.java
VM.java
VMStatic.java
 - Modify [bll](#) vm add and update command insert two properties into ResultSet
AddVmCommand.java
AddVMTemplateCommnad.java
UpdateVmCommand.java
VmManagerCommandBase.java
 - Modify [restapi](#) vm mapper add two properties define
VmMapper.java
 - Modify [vdsbroker](#) vm info builder with two properties
VdsProperties.java
VmInfoBuilder.java
 - Modify [dal](#) vm DAO operation update related insert, update SQL command
VmDAODbFacadeImpl.java
VmStaticDAODbFacadeImpl.java
 - Add [sql_update](#) add two VarChar(100) into vmstatic table
 - Change other configuration files
6. Change vds create process and capabilities with host numa info
- Modify cap collection process use numactl get the host numa info
/vds/vds/cap.py
 - Modify vm create operation rebuild the domain document for libvirt use
/vds/vds/vm.py

Consideration

- User could directly customize the NUMA related tuning on GUI interface

Pros and Cons

	Proposal A	Proposal B
Scene	For advanced user User should know how to setup hook script and modify user custom properties define file on engine	For general user User should know each format of tuning set
Deployment	Every vdsd host should install the hook script rpm and restart the vdsd service will have the feature	No change on host, just install or upgrade the newest version
Code Complex	Simple change	The change effect every module of ovirt engine and node side, and the database should upgrade also
Code Dependency	No dependency for other modules	The ovirt node side should change for first step, after it integrated the engine side will change GUI interface, database, rest, SDK

Issue to be investigated

- How to dealing with live guest migration